Fig. 1

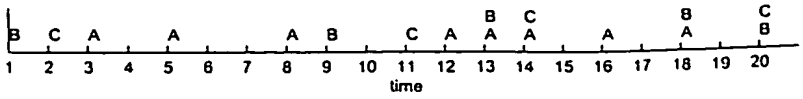


Fig. 2

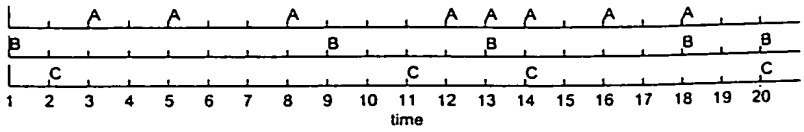


Fig. 3A

Fig. 3B

Fig. 3C

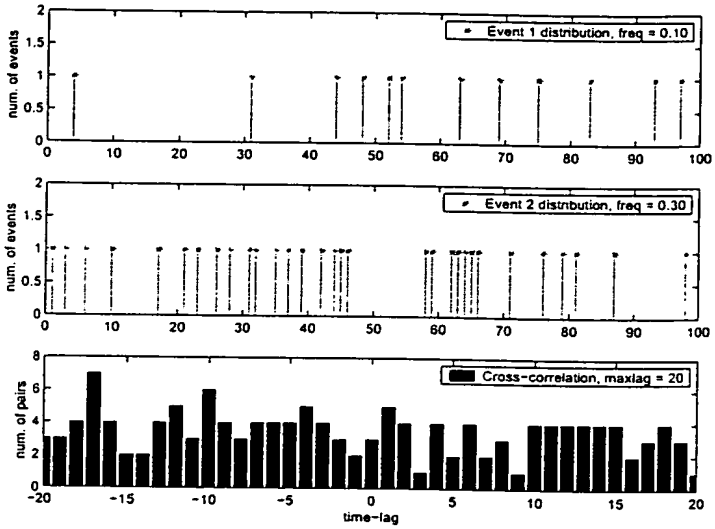


Fig. 4A

Fig. 4B

Fig. 4C

Fig. 4D

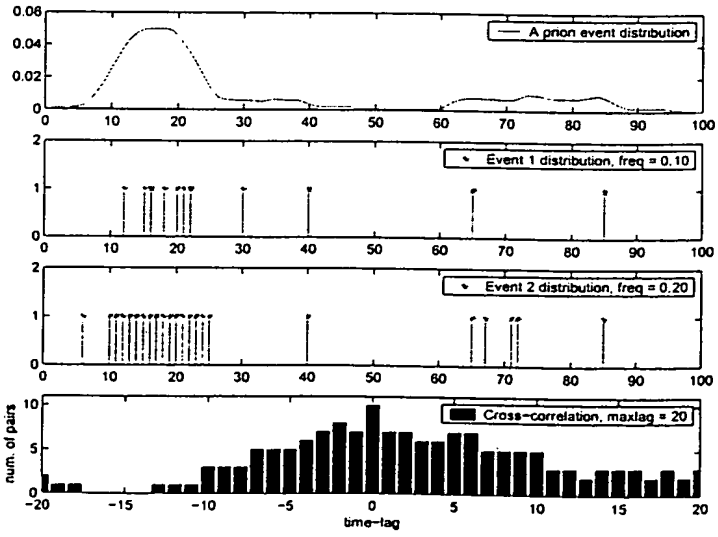
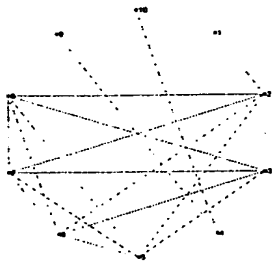
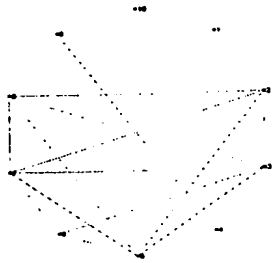


Fig 5A



(a)



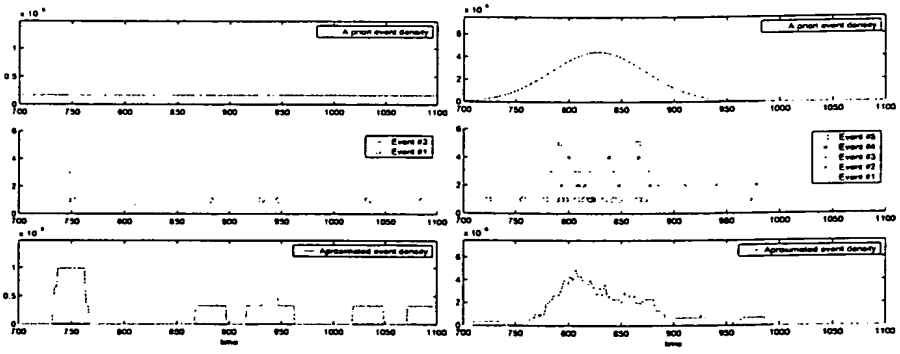
(b)

Fig. 5B

Fig. 6A1

Fig. 6A2

Fig. 6A3



(a) (b)

Fig. 6B1

Fig. 6B2

Fig. 6B3

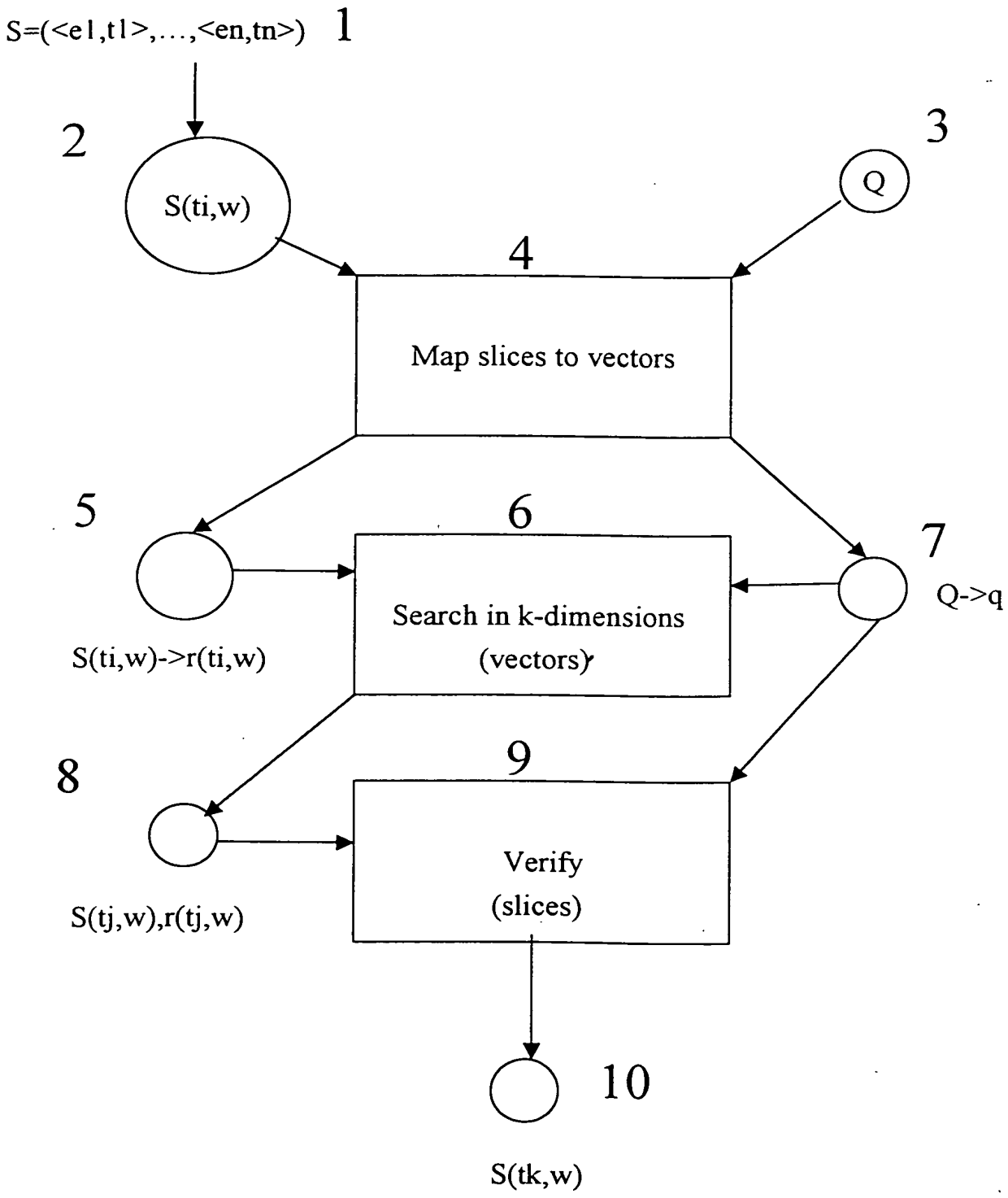

S=(<e1,t1>,...,<en,tn>)
1
2
S(ti,w)
3
Q
4
Map slices to vectors
5
S(ti,w)->r(ti,w)
6
Search in k-dimensions
(vectors)
7
Q->q
8
S(tj,w),r(tj,w)
9
Verify
(slices)
10
S(tk,w)

Figure 7.

Figure 8

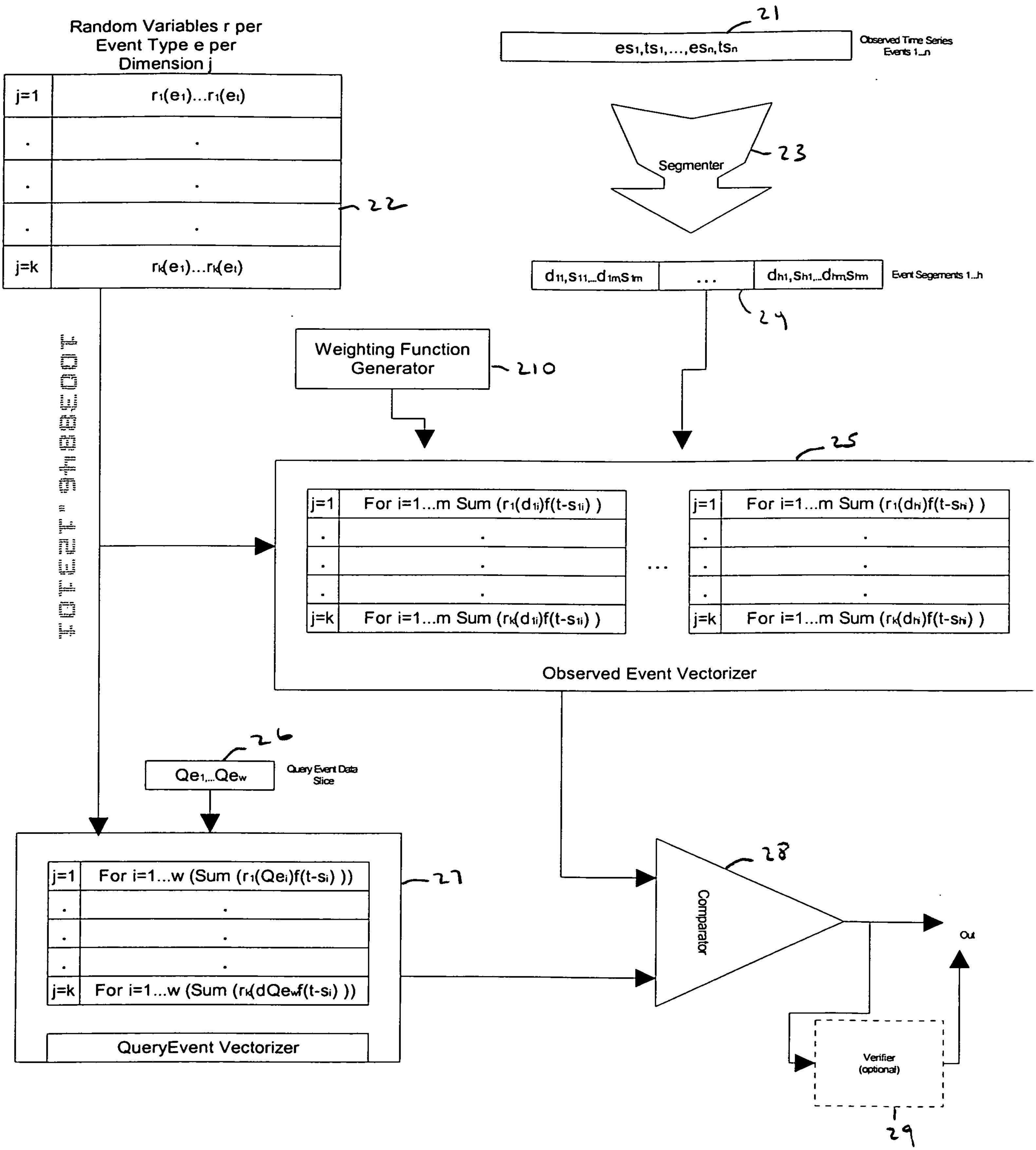

Random Variables r per Event Type e per Dimension j
j=1
r1(e1)...r1(el)
j=k
rk(e1)...rk(el)
22
21
es1,ts1,...,esn,tsn
Observed Time Series Events 1..n
Segmenter
23
d11,S11...d1mS1m
...
dh1,Sh1...dhmShm
Event Segements 1..h
24
Weighting Function Generator
210
25
j=1
For i=1...m Sum (r1(d1i)f(t-s1i))
j=k
For i=1...m Sum (rk(d1i)f(t-s1i))
j=1
For i=1...m Sum (r1(dhi)f(t-shi))
j=k
For i=1...m Sum (rk(dhi)f(t-shi))
Observed Event Vectorizer
26
Qe1,...Qew
Query Event Data Slice
j=1
For i=1...w (Sum (r1(Qei)f(t-si)))
j=k
For i=1...w (Sum (rk(dQewf(t-si)))
27
QueryEvent Vectorizer
Comparator
28
Out
Verifier (optional)
29

Fig. 9

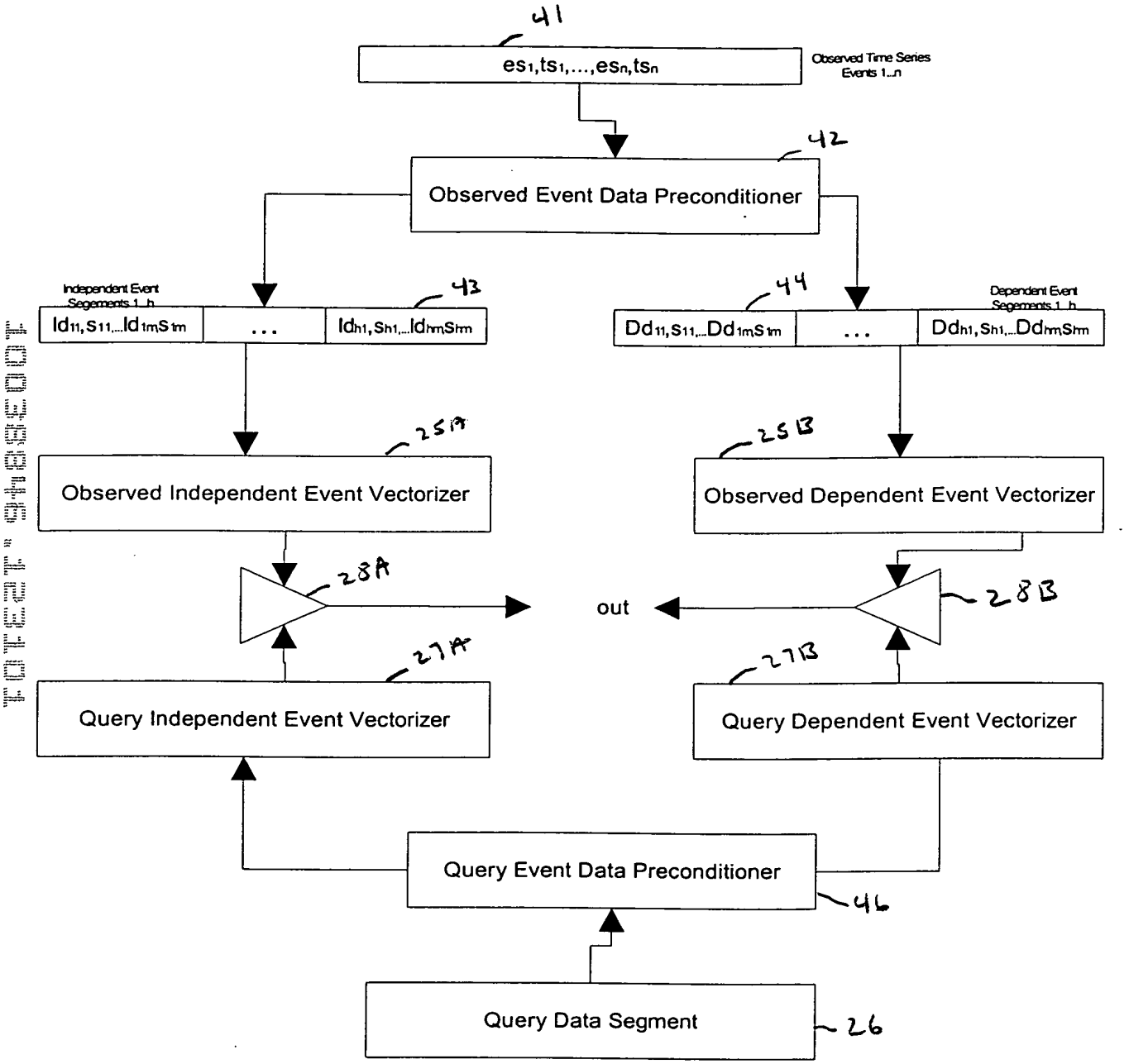

41
es1,ts1,...,esn,tsn
Observed Time Series Events 1..n
42
Observed Event Data Preconditioner
Independent Event Segements 1..h
43
Id11,s11,...Id1mS1m
...
Idh1,sh1,...IdhmShm
44
Dependent Event Segements 1..h
Dd11,s11,...Dd1mS1m
...
Ddh1,sh1,...DdhmShm
25A
Observed Independent Event Vectorizer
25B
Observed Dependent Event Vectorizer
28A
out
28B
27A
Query Independent Event Vectorizer
27B
Query Dependent Event Vectorizer
Query Event Data Preconditioner
46
Query Data Segment
26

Fig. 9A

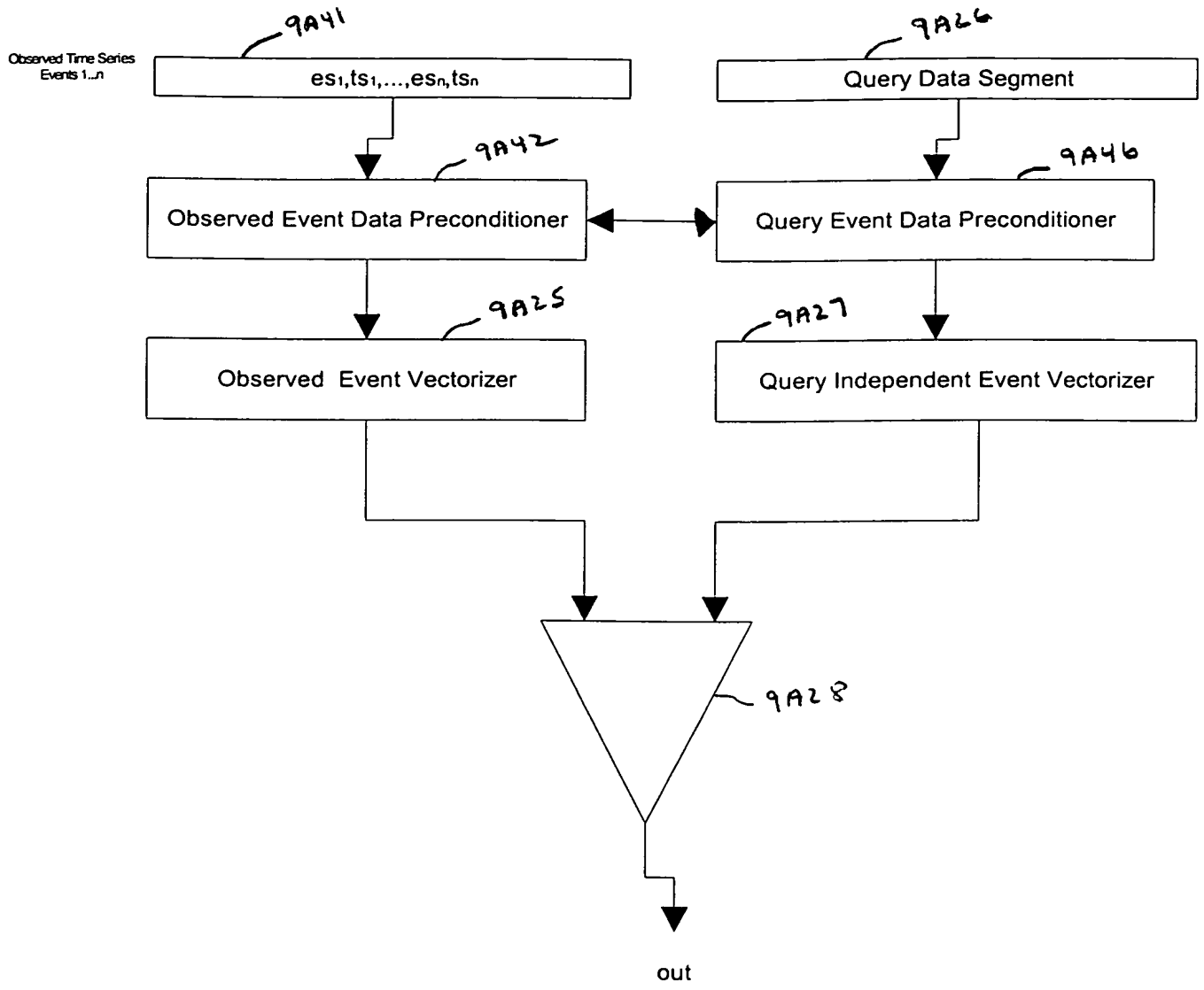
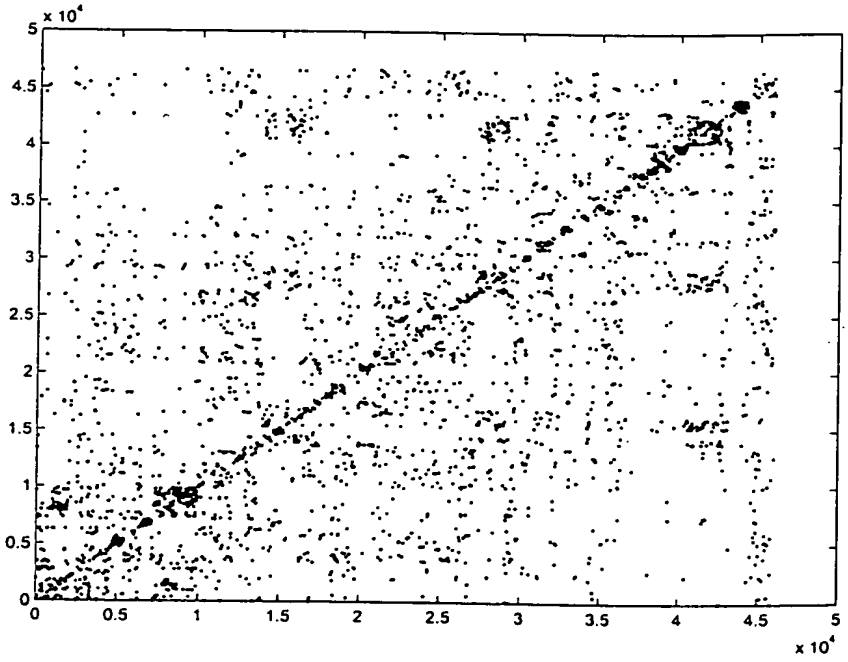

9A41
Observed Time Series Events 1..n
es1,ts1,...,esn,tsn
9A26
Query Data Segment
9A42
Observed Event Data Preconditioner
9A46
Query Event Data Preconditioner
9A25
Observed Event Vectorizer
9A27
Query Independent Event Vectorizer
9A28
out

Fig. 10



The location of the closest slice to the query slice, for every 10th event in the sequence and for W=1000 sec.

Fig. 11

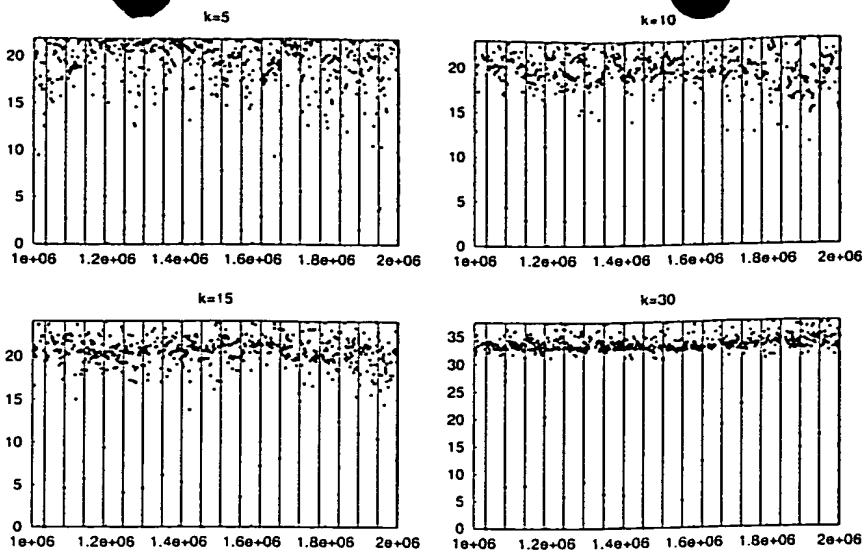
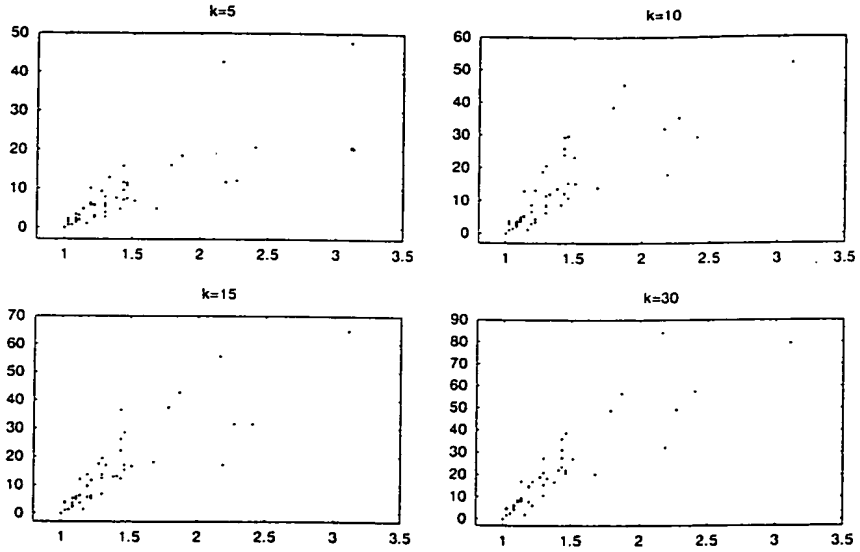


Fig. 12



Distances of target windows against the density ratio, alarm data

Fig. 13

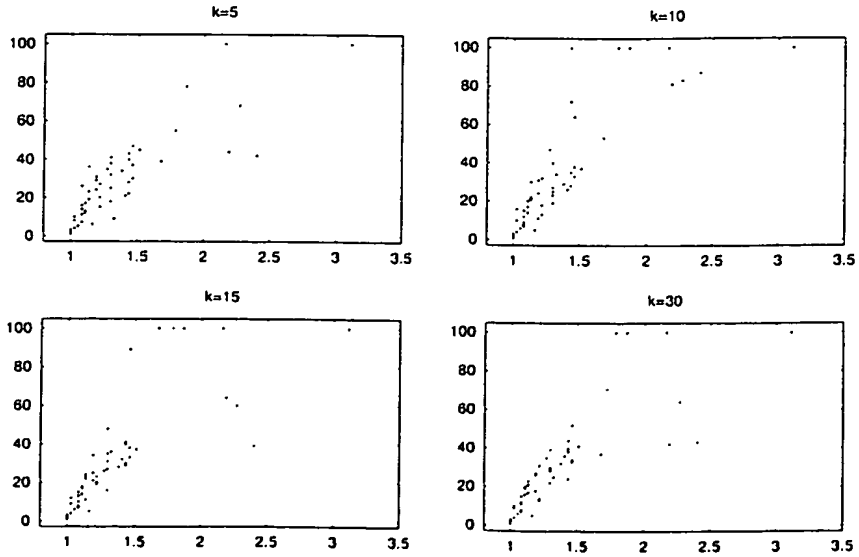


Fig. 14

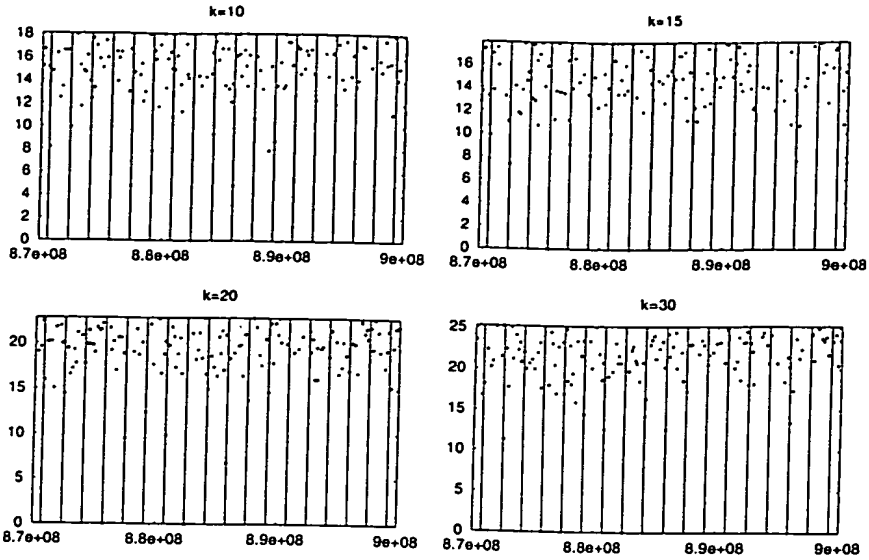


Fig. 17

Dist	Window	Closest	Offset
0.00	1461230	1461230	exact
0.00	2157420	2157420	exact
0.00	1032800	1032800	exact
1.02	2210970	2210970	exact
1.26	497272	497272	exact
1.32	711484	711484	exact
2.10	872143	872143	exact
2.59	3067820	3067820	exact
3.55	1568330	1568330	exact
3.61	2425180	2425180	exact
3.68	604378	604378	exact
4.00	657931	657931	exact
4.04	1247010	1247010	exact
4.55	1300570	1300570	exact
4.57	925696	925696	exact
4.57	3121370	3121370	exact
4.58	1086360	1086360	exact
4.79	2532290	2532290	exact
5.27	1407670	1407670	exact
5.64	2371432	2371630	-198
5.70	1139910	1139910	exact
5.82	2585840	2585840	exact
6.13	1354120	1354120	exact
7.17	1621733	1621880	-147
7.95	1996493	1996760	-267

Dist	Window	Closest	Offset
8.11	1675060	1675440	-380
8.41	3014113	3014260	-147
8.57	2799882	2800050	-168
8.58	979102	979249	-147
9.32	1193557	1193460	97
10.18	818800	818590	210
10.26	2639124	2639390	-266
10.42	1942820	1943200	-380
10.49	2853333	2853600	-267
11.02	2478783	2478730	53
11.02	1889383	1889650	-267
11.33	2103614	2103860	-246
12.17	2692793	2692940	-147
12.41	1835763	1836100	-337
12.91	2906893	2907160	-267
13.14	2264140	2264520	-380
13.75	3059438		missed
13.77	1428734		missed
14.08	2959387		missed
14.17	755127		missed
14.43	1961635		missed
14.59	2053796		missed
14.88	1729345	1728990	355
15.01	1116290		missed
15.26	2087183		missed

Dist=Distance to query window
Window=Position (time) of window found
Closest=Position of closest target window, if closer than 1000
Offset=Difference of this window and closest target

Distances of fifty closest selected windows, $k = 15$, alarm data

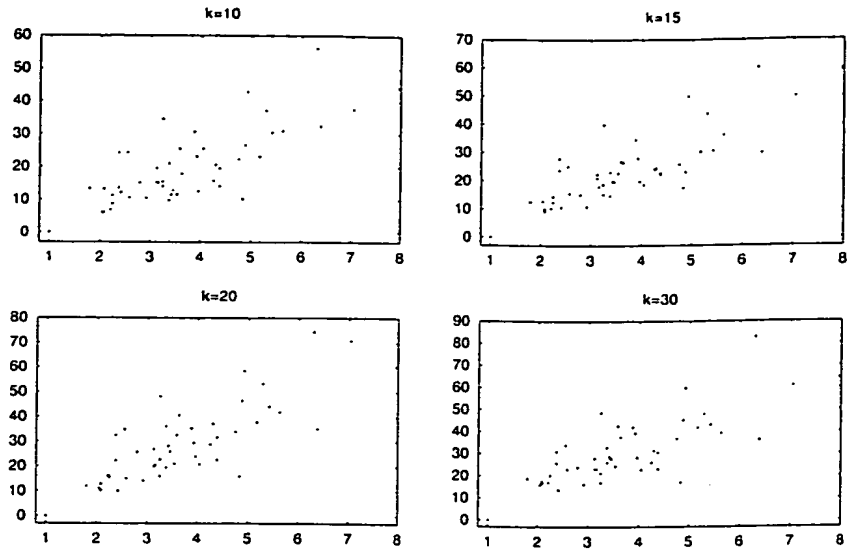


Fig. 15

Distances of target windows against density ratio, Entree Chicago data

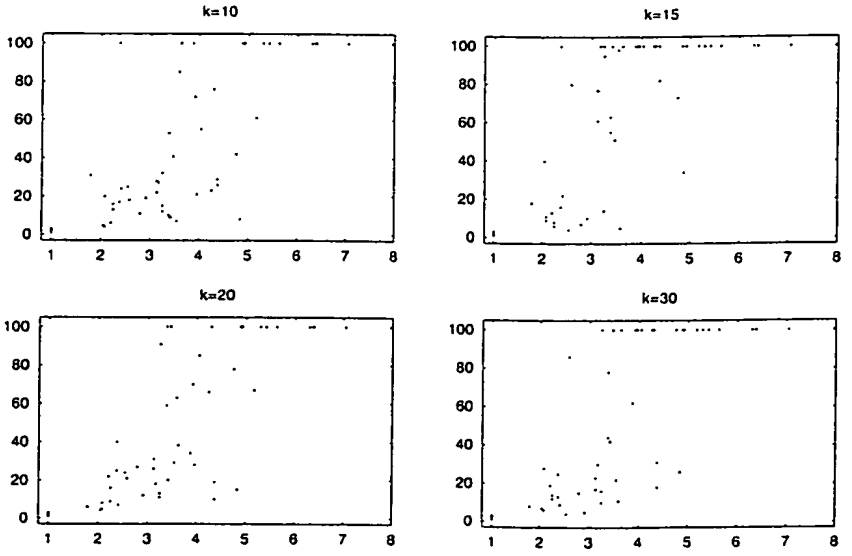


Fig. 16

Ranks of target windows against the density ratio, Entree Chicago data